

58



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/977,516	10/12/2001	David S. Allison	0007056-0203/P5946	3934
32615	7590	07/27/2005	EXAMINER	
OSHA LIANG L.L.P./SUN 1221 MCKINNEY, SUITE 2800 HOUSTON, TX 77010			ROCHE, TRENTON J	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 07/27/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/977,516

Applicant(s)

ALLISON, DAVID S.

Examiner

Trenton J. Roche

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 June 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-45 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-45 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 17 November 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. This office action is responsive to communications filed 9 June 2005.
2. Per applicant's request, amended claims 1, 9, 17, 26 and 38 have been entered. Claims 1-45 are now pending.
3. Claims 1-45 have been examined.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Java!" by Ritchey in view of "Object-Oriented Programming and the Objective-C Language" by NeXT Software, Inc., hereafter referred to as Objective-C.

Per claim 1:

Ritchey discloses:

- a method for instantiating an object, comprising determining an object type of said object
(“This statement tells the compiler to create a variable with the name *identifier* of type *type*” on page 138, section titled Declaring Variables)
- reserving a memory block on a memory structure, the size of said memory block being determined according to said object type, and said memory structure being selected

according to said object type (Note pages 134, 135 and 139. The sections describe one example of an object type, the Integer. Further, it is disclosed that a specific amount of memory is allocated depending on the object type, be it integer, long integer, floating-point, etc. "The int type integer is 32 bits long..." on page 139. Further, pages 339-341 disclose when certain memory structures, specifically that of the stack and heap, are selected and used.

- Creating a reference structure to said object (Note page 131, section titled Identifiers, and further, "This statement tells the compiler to create a variable with the name *identifier* of type *type*" on page 138, section titled Declaring Variables. The identifier is the reference structure of the object.)

substantially as claimed. Ritchey does not explicitly disclose the object being written in a dynamically typed language. Objective-C discloses the ability to write objects in a dynamically typed language, and discusses advantages of utilizing a dynamically typed language as opposed to a static typed language. (Note section titled "Dynamic Typing" on pages 13 and 14 of Chapter 1 of Objective-C). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the dynamic typing ability disclosed by Objective-C into the language of Ritchey. Objective-C discloses that in a statically typed language, it is impossible to let run-time factors influence the decision about what kind of object should be assigned to a variable, and so the methods executed and the results returned are always fixed. However, dynamically typing the objects would provide the added ability to wait until run time to discover the class of a variable, thereby allowing different versions of methods to be executed and different results to be returned, depending on the class of the receiver, as noted in Objective-C on pages 13 and 14 of Chapter 1.

Art Unit: 2193

Per claim 2:

The rejection of claim 1 is incorporated, and further, Ritchey discloses obtaining a keyword, and identifying said keyword as claimed (“This statement tells the compiler to create a variable with the name *identifier* of type *type*” on page 138, section titled Declaring Variables. The compiler obtains the type designation, which is a keyword, and identifies it as a certain keyword. Further, note page 132, section titled Keywords, and the associated table 6.3.)

Per claim 3:

The rejection of claim 2 is incorporated, and further, Ritchey discloses executing a second of constructor statements if said set contains at least one statement as claimed (Note page 132, section titled Keywords, and the associated table 6.3. *class* is a designated keyword, and further, note pages 172 and 173, section titled Object Creation and Destruction, wherein a class Rectangle is declared which contains a constructor with statements in the constructor which would be executed when the class is instantiated.)

Per claim 4:

The rejection of claim 3 is incorporated, and further, Ritchey discloses a keyword identifying said object type as a class as claimed (Note page 132, section titled Keywords, and the associated table 6.3. *class* is a designated keyword.)

Per claim 5:

Art Unit: 2193

The rejection of claim 4 is incorporated, and further, Ritchey discloses a memory structure being a heap as claimed (Note page 341, section titled The Garbage Collected Heap, wherein “The heap is the store of memory from which class instances are allocated”)

Per claim 6:

The rejection of claim 3 is incorporated, and further, Ritchey discloses a keyword identifying said object type as a function as claimed (Note page 168, section titled Methods in Java Classes. The structure of the method declaration, along with the associated returnType keyword, tells the compiler that the object is a function.)

Per claim 7:

The rejection of claim 6 is incorporated, and further, Ritchey discloses optionally returning a value to a calling statement (“Every method must return a value or be declared as void” on page 168, section titled Methods in Java Classes), deleting said reference structure and freeing said memory block as claimed (“When Java sees there are no more references to an object, it places it on the stack for garbage collection” on page 43, section titled Memory Management and Threads)

Per claim 8:

The rejection of claim 7 is incorporated, and further, Ritchey discloses a memory structure being a stack (Note page 339, section titled The Java Stack)

Per claims 9-16:

Art Unit: 2193

Claims 9-16 are directed to a computer program product for performing the methods of claims 1-8, respectively, and are rejected for the reasons set forth in connection with claims 1-8, respectively.

Per claim 17:

Note the rejection regarding claim 1. In Java, the interpreter is what inherently determines object types, which in turn calls the inherent storage allocation subsystem for controlling memory allocation, and the system inherently includes an access control subsystem for accessing the identifier reference structure.

Per claim 18:

The rejection of claim 17 is incorporated, and further, Ritchey discloses the interpreter comprising a lexical analyzer, and a semantic parser, where said analyzer is configured so as to pass tokens representing keywords to said parser, and said parser is configured so as to identify said tokens as claimed (Note page 129, section titled The Java Token Set. The system inherently has a lexical analyzer and a semantic parser.)

Per claim 19:

The rejection of claim 18 is incorporated, and further, Ritchey discloses a statement execution subsystem, said execution subsystem configured so as to automatically execute a set of constructor statements as claimed (Note the rejection regarding claim 3)

Per claim 20:

The rejection of claim 19 is incorporated, and further, note the rejection regarding claim 4.

Art Unit: 2193

Per claim 21:

The rejection of claim 20 is incorporated, and further, note the rejection regarding claim 5.

Per claim 22:

The rejection of claim 19 is incorporated, and further, note the rejection regarding claim 6.

Per claims 23:

The rejection of claim 22 is incorporated, and further, note the rejection regarding claim 7.

Per claim 24:

The rejection of claim 23 is incorporated, and further, note the rejection regarding claim 7. Further, note page 341, section titled The Garbage Collected Heap. The garbage collection is done automatically in Java, and after the statements in the constructor have been executed, the memory used by the constructor class would be automatically deleted.

Per claim 25:

The rejection of claim 24 is incorporated, and further, note the rejection regarding claim 8.

Per claims 26-34:

Claims 26-34 are directed to an object instantiation component for performing the steps of the system disclosed in claims 17-25, respectively, and are rejected for the reasons set forth in connection with claims 17-25, respectively.

Per claims 35-37:

The rejections of claims 9, 17 and 26 are incorporated, respectively, and further, Ritchey discloses the object including a definition of one or more objects as claimed (Note section titled “Classes in Java,” page 166 of Ritchey. “Note the beginning and ending curly braces, which designate the limits of the class definition.” Further, note the definition of the Rectangle class on page 169, wherein the class object includes a definition of one or more objects, specifically ints.)

Per claim 38:

Ritchey discloses:

- a method for instantiating an object, comprising determining an object type of said object (“This statement tells the compiler to create a variable with the name *identifier* of type *type*” on page 138, section titled Declaring Variables)
- reserving a memory block on a memory structure, the size of said memory block being determined according to said object type, and said memory structure being selected according to said object type (Note pages 134, 135 and 139. The sections describe one example of an object type, the Integer. Further, it is disclosed that a specific amount of memory is allocated depending on the object type, be it integer, long integer, floating-point, etc. “The int type integer is 32 bits long...” on page 139. Further, pages 339-341 disclose when certain memory structures, specifically that of the stack and heap, are selected and used.
- Creating a reference structure to said object (Note page 131, section titled Identifiers, and further, “This statement tells the compiler to create a variable with the name *identifier* of type

Art Unit: 2193

type” on page 138, section titled Declaring Variables. The identifier is the reference structure of the object.)

- the object including a definition of one or more objects (Note section titled “Classes in Java,” page 166 of Ritchey. “Note the beginning and ending curly braces, which designate the limits of the class definition.” Further, note the definition of the Rectangle class on page 169, wherein the class object includes a definition of one or more objects, specifically ints.)

substantially as claimed. Ritchey does not explicitly disclose the object being written in a dynamically typed language, and does not explicitly disclose the object including a definition of one or more objects of the same object type. Objective-C discloses the ability to write objects in a dynamically typed language, and discusses advantages of utilizing a dynamically typed language as opposed to a static typed language, and that ability for an object to have definitions of the same object type. (Note section titled “Dynamic Typing” on pages 13 and 14 of Chapter 1 of Objective-C, and further, note section “Id” and “Dynamic Typing” on pages 2 and 3 of Chapter 2. The Id object, which is nonrestrictive, is an object wherein the type is discovered at runtime. Furthermore, note section “Class Objects” on pages 10 and 11 of Chapter 2, wherein it is disclosed that Class objects are full-fledged objects that can be dynamically typed, and as such, can contain definition of other ids or classes). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the dynamic typing ability disclosed by Objective-C into the language of Ritchey. Objective-C discloses that in a statically typed language, it is impossible to let run-time factors influence the decision about what kind of object should be assigned to a variable, and so the methods executed and the results returned are always fixed. However, dynamically typing the objects would provide the added ability to wait until run time to discover the class of a variable, thereby

Art Unit: 2193

allowing different versions of methods to be executed and different results to be returned, depending on the class of the receiver, as noted in Objective-C on pages 13 and 14 of Chapter 1.

Per claims 39:

The rejection of claim 38 is incorporated, and further, note the rejection regarding claim 2.

Per claims 40:

The rejection of claim 39 is incorporated, and further, note the rejection regarding claim 3.

Per claims 41:

The rejection of claim 40 is incorporated, and further, note the rejection regarding claim 4.

Per claims 42:

The rejection of claim 41 is incorporated, and further, note the rejection regarding claim 5.

Per claims 43:

The rejection of claim 40 is incorporated, and further, note the rejection regarding claim 6.

Per claims 44:

The rejection of claim 43 is incorporated, and further, note the rejection regarding claim 7.

Per claims 45:

The rejection of claim 44 is incorporated, and further, note the rejection regarding claim 8.

Response to Arguments

6. Applicant's arguments filed 5 April 2005 have been fully considered but they are not persuasive.

Per claims 1-45:

7. The applicant states that Ritchey does not teach or suggest the instantiation of an object, wherein the object is written in a dynamically typed language. Applicant's arguments with respect to claims 1-45 have been considered but are moot in view of the new ground(s) of rejection. The combination of Ritchey and Objective-C discloses the required limitations. Furthermore, it is noted that the current claims, as amended, do not impart how the object being written in a dynamically typed language results in any sort of required modification of the methods employed by statically typed languages. The claimed limitations, interpreted according to the broadest reasonable interpretation, do not clearly describe how the object being written in a dynamically typed language requires substantially different steps than if the object were being written in a statically typed language. As Ritchey does not necessarily preclude the ability to dynamically type objects, the combination would at least be obvious in view of the advantages known in the art concerning dynamically typed languages. For these reasons, the rejection regarding claims 1-45 is proper and maintained.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Art Unit: 2193


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Trenton J. Roche whose telephone number is (571) 272-3733. The examiner can normally be reached on Monday - Friday, 9:00 am - 5:30 pm.

9. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Trenton J Roche
Examiner
Art Unit 2193

TJR


ANIL KHATRI
PRIMARY EXAMINER